

Replace the paragraph on page 6, lines 9-10, with the following paragraph:

A2
FIG. 8B illustrates a network of databases according to another embodiment of the invention.

On page 6, after line 10, **insert** the following paragraph:

A3
FIG. 8C illustrates a network of databases according to yet another embodiment of the invention.

Replace the paragraph on page 6, lines 15-16, with the following paragraph:

A4
FIG. 11A is an illustration of an update database record in an update database for a database.

On page 6, after line 16, **insert** the following paragraphs:

A5
FIG. 11B is an illustration of an update search-routing database record according to an embodiment of the invention.

FIG. 12 is a flow chart illustrating a record update process according to an embodiment of the invention.

Replace the paragraph on page 15, lines 5-13 with the following paragraphs:

A6
In the example illustrated in FIG. 8A, the New York office 201 includes at least four databases: New York database 205 and server 208 "Larry" having informational records associated with New York; a New Jersey database 206 and database server 209 "Moe" having information records associated with New Jersey; and an update database 207 and database server "Curly" having records which have information to update particular records in the New York or New Jersey databases. Upon receipt of requests from a client, these database servers search the databases for selected records and pass them back over the network. A plurality of client terminals 213, 214 are provided through which requests are received and fulfilled.

Replace the paragraph on page 15, lines 21-27 with the following paragraph:

A7
In the example shown in FIG. 8A, the Los Angeles office 202 has a CA database 215 database server "Kato" 216. Like the New York office, the Los Angeles office also has a search routing database 217 and proxy server "Hank" 218. The San Francisco office 203 in this example only has a search routing database 230 and a proxy server "Oscar" 231. The Seattle office has only a Washington database 234 and database server "Joe" 235. As with the New York office, each of the Los Angeles, San Francisco and Seattle offices have client terminals 219, 220, 232, 233, 236, 237 through which requests are received and fulfilled.

Replace the paragraphs on page 17, lines 8-32 with the following paragraphs:

AS
An example of a set of search routing tables using encapsulated complexity is illustrated in FIG. 8A. As explained in connection with FIG. 4, applying the modified search to a search routing table, a route is identified for the original search request through the database network. Examples of search routing tables corresponding to the system illustrated in FIG. 8A are provided. Table 251 is an example of a search routing table for the New York office 201. Table 251 would be maintained on search routing database 211. Table 252 is an example of a search routing table for the Los Angeles office 202. Table 252 would be maintained on search routing database 217 in the Los Angeles office 202. Table 253 is an example of a search routing table for the San Francisco office 203 and would be maintained on search routing database 230 in the San Francisco office 203. The Seattle office 204, without a search routing database or proxy server, would not have a corresponding search routing table.

Suppose a search request is entered via client terminal one 232 in San Francisco for a New York telephone number for a café named "Joe's Café." Because a state is supplied with the request, a modified search request is generated by the system with the state field filled in with the data "NY." Proxy server "Oscar" 231 would receive the modified request and search search-routing database 230 for responsive routing directions. Entry 260 in search routing table 253 in search routing database 230 would return "Stooges" database as the appropriate database to search for data responsive to the modified request.

AG The system would then send over the WAN the original search request to the Stooges proxy server 212 in the New York office 201. The Stooges proxy server in the New York office does not know the difference between a search request generated at a remote site transmitted over the WAN versus the same original search request generated by a local user at client terminal one 213. Thus, the Stooges proxy server 212 will process the original search request received from the San Francisco office 203 just as if it had received the request via client terminal one 213.

Replace the paragraphs on page 18, lines 13-32 with the following paragraphs:

AG Certain offices may not include the capability to search remotely. Referring to the example shown in FIG. 8A, the Seattle office 204 does not include a proxy server or search routing database. Searches received at the Seattle office 204 may only be satisfied by conducting a search on the Washington database 239. The other three offices, New York, Los Angeles and San Francisco, also have the capability to search the Washington database 239.

In addition to the search routing tables in each of the search routing databases 211, 217 and 230, there is defined for each proxy server one or more default routes. Default routes specify how a search request is routed when a modified search request fails to return a route. This typically occurs when either (i) the search request does not include a field that is used for routing; (ii) the search request includes a specified routing field but no data is populating that field; or (iii) there is data in the routing field(s) that is not matched in the corresponding field(s) in the routing database. For example, suppose a search request is received into the system via the Los Angeles Office 202 (FIG. 8A). If this search request does not have a state field or has a state field but does not have a state specified, the search routing database would return a default route. Generally, the default route is defined to route the request to the data that is available to the servers that are on the local host, in this example, CA database "Kato" 216.

On page 18, after line 32, **insert** the following paragraph:

A10
FIGs. 8B and 8C illustrate further embodiments of database network architectures for performing the query searches, database access and record updates described herein. In Fig. 8C, an index "U" is used to distinguish an update database record from its pre-update counterpart.

Replace the paragraph on page 19, lines 1-14 with the following paragraph:

A11
A flow chart illustrating request processing flow is provided in FIG. 10. The flow chart illustrates how a request is accepted and sent to the appropriate database. At step 261, a search request is received from a client. The search request is inspected to determine whether it includes a defined "route-by" field(s) (step 262). Although the present invention has been described using the state as the "route-by" field, it is understood to one skilled in the art that any data element or combination of data elements may be used. If the search request does not have a defined "route-by" field or such a field is unpopulated (step 263), then a default route is identified (step 264) and the original request is submitted to the identified default route (step 265). If the search request has a populated route-by field (step 263), a search of the search routing database is performed using defined "route-by" fields with the content of the corresponding field(s) in the original request (step 266). Because this is often a subset of the entire original request, it is often referred to as a "modified search request." If the search of the search routing database returns one or more database identifiers, the original request is submitted to the identified databases for fulfillment (step 267).

Replace the paragraphs on page 20, line 18 - page 21, line 21 with the following paragraphs:

A12
As illustrated in FIG. 8A, an update database is created that corresponds to one or more main databases that the system wants to update (for example, the Curly database has records that are adds and deletes corresponding to records in the Larry and Moe main databases). The system administrators set up the search routing tables such that the update databases are placed on the same routes as the main databases to which they correspond. When a main

database is a default route, the update database corresponding to that main database is made part of that default route. Similarly, when a main database is part of a specific route identified in a search routing table, the corresponding update database occurs on the same route. Referring to FIG. 9B, it is noted that any search to database 28, 30, 32 is also routed to update database 28u, 30u and 32u. Turning to the databases shown in Fig. 8A, any search to the New York database is routed to both the main New York database Larry and the update database Curly. Similarly, any search to the New Jersey database is routed to both the main New Jersey database Moe and update database Curly. As a result, a search of the New York database will return both results from the New York database as well as update database.

A12 Updating of records from the search results from the New York and update databases is a further extension to the merge algorithm with duplicate removal. Update is a method defined by which a record can be identified for removal or addition. According to the preferred embodiment, this method can be a defined value in a defined field. FIG. 11A shows an exemplary update database record 34u having a plurality of fields 52u, each field 52u having an associated field identifier 54u and potentially containing data 56u. Similar to the record illustrated in FIG. 2, the update record has six fields 52u for storing data 56u regarding an individual's last name, first name, middle initial; city, state, and telephone number. Update record 26u, 34u contains an additional field that is used by the present invention to perform on-the-fly updating. As depicted in FIG. 10, this additional field is identified as UPDATE_ACTION 201u. The UPDATE_ACTION field 201u is populated with a predefined value. According to the preferred embodiment, when the UPDATE_ACTION field contains a "D" then records are to be deleted. Any other value in the UPDATE_ACTION field constitutes an insertion. It is readily apparent to one of ordinary skill in the art that any appropriate data structure for the UPDATE_ACTION field and any particular choice of values in the UPDATE_ACTION field could be used in accordance with the principles of this invention.

The present invention accomplishes the appearance of record deletion by implementing an algorithm using a comparison of a database record with an update database record. This technique is shown with reference to Fig. 12. As shown by Fig. 12, in one

A12
embodiment, a query search is performed in the same or similar manner as shown in Fig. 5 (steps 70-82). After the identified databases are searched using the original search requests (step 82), the identified update database(s) are searched using the original search request (step. 284). The search results are then returned to the proxy server (step 286). The proxy server assures that in any case where there are one or more database records which are logically duplicate to a update database record (according to the defined matching rules) AND the update database record also includes the defined delete indicator (UPDATE_ACTION = D) (step 288), then NONE of these records are written to the output (step 290).

Replace the paragraphs on page 22, lines 1-14 with the following paragraphs:

A13
Where there are no more records present in the input buffers that match the record in the output buffer, the output mode flag is checked. If it indicates "delete," then the record in the output buffer is discarded and no record is written to the output. If the flag does not indicate "delete", then the record in the output buffer is written to the output (step 292) and the buffer and flag are cleared.

The present invention encompasses a number of additional embodiments and capabilities. One further embodiment of the present invention provides for direct database access on a local host without database servers. More particularly, this embodiment has the proxy server controlling the local databases directly, thus fulfilling the role of database server as well as proxy server. This embodiment takes advantage of the fact that the proxy server generally provides a fully inclusive superset of the functionality of the database server. This particular embodiment is not illustrated in FIG. 8A but would have the Stooges database interfacing directly with the New York, New Jersey and update database, thereby eliminating the Larry, Moe and Curly database servers.

IN THE CLAIMS

Enter amended claims 1-35 as follows:

A14 SUB 1. (Amended) A method of updating database records comprising: